CipherSchool

Assignment-2

choose the correct option

① Local Variable are stored in an area called
Answer → Stack

② when the inheritance is private the private methods in base class are ——— in derived class

Answer → Inaccessible (In C++)

③ State true or false
type conversion is automatic whereas type casting is explicit
Answer → true;

Short answer

① New Keywords → The new operator is which denotes a request for memory allocation on the Heap if sufficient memory is available New

code →
```
#include<iostream>
using name space std;
class car
{
String name;
int num;
```

Date: _____ Page: _____

```
public:
    car(string a, int n)
    {
    cout << " constructor called" <<endl;
      this->name = a;
      this-> num = n;
    }
    void enter()
    {
     cin >> name;
     cin >> num;
    }
    void display ()
    {
    cout << "name:" << name << endl;
     cout << "num:" << num << endl;
    }
};
int main()
{
    car * p = new car ("Honda", 2020);
    p-> display ()
```

output → constructor called

delete keywords → Delete is an operator that is used to destroy array and non-array (pointer) object which are created by new expression.

Code →

```
int main ()
{
int * array = new int [10];
delete [] array;
return 0;
}
```

(1) constructors → Constructors is a member function of a class which initializes objects of a class in C++, constructors is automatically called when object create

their are 3 type of constructors.

① Default ② Parameterized ③ Copy.

↓

Ex.

Date: _____ Page: _____

(3) procedural oriented programming	object oriented programming
(1) In procedural pr. program is divided into small parts called function	(1) in object oriented pro. program is divide into small parts. calle objects.
(2) Procedural pro. follows top down approach	(2) object oriented pro. follows bottom up approach
(3) There is no access specifier in procedural pro.	(3) object oriented pro. have access specifies like private, pub
(4) Example → C, Fortran, pascal Bas	(4) ~~Ex. c++~~, java, Python C# etc.

Long Answer type

(1) polymorphism is a feature of OOPs that allows the object to behave differently indifferent condition. In c++ we have two type of polymorphism.

(A) compile time polymorphism → function overloading and operator overloading are perfect example of Compile time polymorphism

```
#include<iostream>
using namespace std;

class Add{
public:
    int sum (int num1, int num2) {
        return num1+num2;
    }
    int sum (int num1, int num2, int num3) {
        return num1+num2+num3;
    };
int main () {
    Add obj;
    cout << "output:" << obj.sum(10,20) << endl;
    cout << "output:" << obj.sum(11,22,33);
    return 0;
}
```

output: 30
output: 66.

(B) Runtime Polymorphism → function overRiding main example of Runtime Polymorphism.

```
Code →  #include <iostream>
        using namespace std;
        class A2
        public:
          void disp () {
            cout << "super class function" << endl;
```

Date: _____ Page: _____

```
};
Class B: public A {
public:
    void disp() {
        cout << "sub class fuction";
    }
};
int main()
{
    Aobj;
    obj.disap();
    B obj2;
    obj.disp();
    return 0;
}
```

Output

superclass function
subclass fuction.

Date: _____ Page: _____

(2)

```
#include <bits/stdc++.h>
using namespace std;

void sort012(int a[], int arr_size)
{
  int lo = 0;
  int hi = arr_size - 1;
  int mid = 0;
while (mid <= hi) {
  switch (a[mid]) {
  case 0:
    swap (a[lo++], a[mid++]);
    break;
case 1
   mid ++
    break;
case 2
Swap (a[mid], a[hi--]);
   break
}
}
}
void printarray (int arr[], int arr_size)
{
for (int i = 0; i < arr_size; i++)
    cout << arr[i] << " ";
}
int main()
```

Date: Page:

```
int arr[] = {0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1};
int n = sizeof(arr) / sizeof(arr[0]);
sort012(arr, n);
cout << "array after segregation ";
printArray(arr, n);
return 0;
}
```

Output

0 0 0 1 1 1 2 2 2 2